Locking and unlocking records

*Maciej Szymczak*

*Version 2024.04.05*

[Functional description 1](#_Toc163232871)

[Technical description 2](#_Toc163232872)

[How was it implemented? 2](#_Toc163232873)

|  |
| --- |
| This enhancement works fine, however Salesforce created a standard solution for locking and unlocking records by flow. See https://help.salesforce.com/s/articleView?id=sf.flow\_ref\_elements\_actions\_lockrecord.htm&type=5 for more. |

# Functional description

This customization provides apex classes that can be executed by flow so that you can build the Quick Action available in Lightning that locks or unlocks the record.

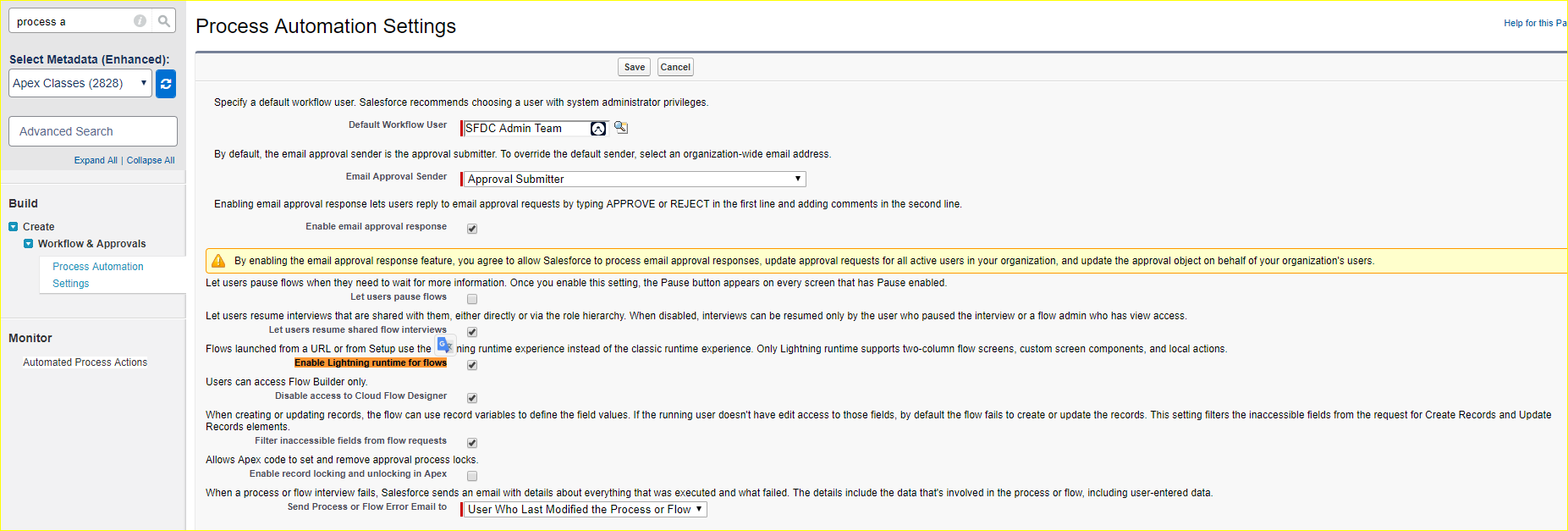
Why do we need it?

In Salesforce Classic there is a button . The button unlocks the Opportunity.

This functionality is not available in Lightning, there is open idea on this here: <https://success.salesforce.com/ideaView?id=0873A0000003XR6QAM>.

Notes:

1. Mark the checkbox **Enable record locking and unlocking in Apex** as true and save in Setup > Create > Workflows and Approvals > Process Automation Settings
2. Optionally: Select this checkbox



It causes flows are rendered in Lighting style.

1. Build the Weblink or Quick action

* In hybrid environment (classic and lightning)

Create and add the **Weblink** to page layouts

* In Lighting environment

Created and the **Quick Action** to page layouts

# Technical description

Add apex action to your flow and select ApprovalIsLocked, ApprovalLock or ApprovalUnlock.

A blue square with white arrow and black text

Description automatically generated

A screenshot of a computer

Description automatically generated

## How was it implemented?

Apex classes were created. Classes expose methods available in flows.

public class ApprovalIsLocked {

/\*

\* This apex class is used by flows

\* @author Maciej Szymczak

\* @version 2019.08.16

\*/

@InvocableMethod

public static List<Boolean> Run(List<String> Ids) {

List<Boolean> result = new List<Boolean>();

for (String p : Ids) {

result.add( Approval.isLocked (p));

}

return result;

}

}